IN THE ABSTRACT:

Please substitute the following annotated abstract for the originally filed abstract:

~~A method and system are provided for enabling scheduling thread execution in a computer system. Initially, a circular array structure is maintained having a plurality of time slots therein, wherein each of the plurality of time slots corresponds to a timeslice during which CPU resources are allocated to a particular thread. Next, each of the time slots in the circular array are configured to include a queue of threads scheduled for execution during that time slot. A pointer index is maintained for referencing one time slot in the circular array and whereby advancement through the circular array is provided by advancing the pointer index. An array of threads requesting immediate CPU resource allocation is also maintained. In operation, a currently executing thread is suspended. Next, a next time slot during which the currently executing thread should next resume execution is calculated. The suspended currently executing thread is then appended to the queue of threads scheduled for execution at the calculated time slot. A next sequential non-empty time slot is identified and the pointer index is updated to point to the identified next sequential non-empty time slot. Any contents of the indexed time slot is then appended to the array of threads requesting immediate CPU resource allocation. The thread at the top of the array of threads requesting immediate CPU resource allocation is then removed and activated.~~ Described are various embodiments for scheduling threads for a multi-tasking operating system. In accordance with some embodiments, a rate-based scheduling algorithm is incorporated to provide a flexible set of Quality of Service guarantees controlling the allocation of CPU to a thread, together with a mechanism to provide latency control for hard-real-time software. Additionally, the present invention is particularly suitable for use in systems where essentially all software runs under the control of the scheduler, from "interrupt handler" functions, though multimedia applications to simple "console" functions and non-real-time tasks. In an additional embodiment, the system and method of the present invention also actively limits the CPU time allocated to a given software thread to defend against malicious or accidental denial of service attacks.